

Dispositivo para detecção de vagas em estacionamento com IOT



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

Sumário

- Introdução
- Trabalhos relacionados
- Hardwares
- Tecnologias de comunicação
- Protótipo
- Resultados
- Conclusão
- Referências

Introdução

- **Problemática: Busca por vagas em estacionamentos.**
 - Tempo gasto em busca por vagas.
 - Congestionamento.
 - Emissão de CO2

Introdução

- **Solução: Sensores de estacionamento**
 - Redução no tempo de procura por vagas
 - Redução dos níveis de CO₂.
 - Redução no consumo de energia.
 - Redução de tráfego.

Introdução



Introdução

- **Construção de um dispositivo IOT para detecção de vagas.**
 - Identificar presença ou não de carros em vagas.
 - Enviar dados para broker.
 - Mostrar em uma aplicação web a disponibilidade das vagas.

Trabalhos relacionados

- **Artigo: “IoT-Enabled Vacant Parking Slot Detection System Using Inkjet-Printed RFID Tags”**
- **Autores(2023)**
 - Lubna (Student Member, IEEE)
 - Adnan Zahid (Member, IEEE)
 - Naveed Mufti (Member, IEEE)
 - Sadiq Ullah (Senior Member, IEEE)
 - Muhammad Waqas Nawaz (Member, IEEE)
 - Abubakar Sharif (Member, IEEE)
 - Muhammad Ali Imran (Fellow, IEEE)
 - Qammer H. Abbasi (Senior Member, IEEE)
- O sistema utiliza sensores RFID para identificar e localizar vagas de estacionamento ocupadas por meio de etiquetas RFID únicas.

Trabalhos relacionados

- **TCC : Projeto de estacionamento inteligente utilizando parking IOT sensor**
- **Autor: Vitor Emmanuel Schoepping (2022)**
- Propõe um dispositivo de detecção de vagas de estacionamento utilizando sensor ToF e uma plataforma de apresentação de dados chamada TAGO.IO.

Trabalhos relacionados

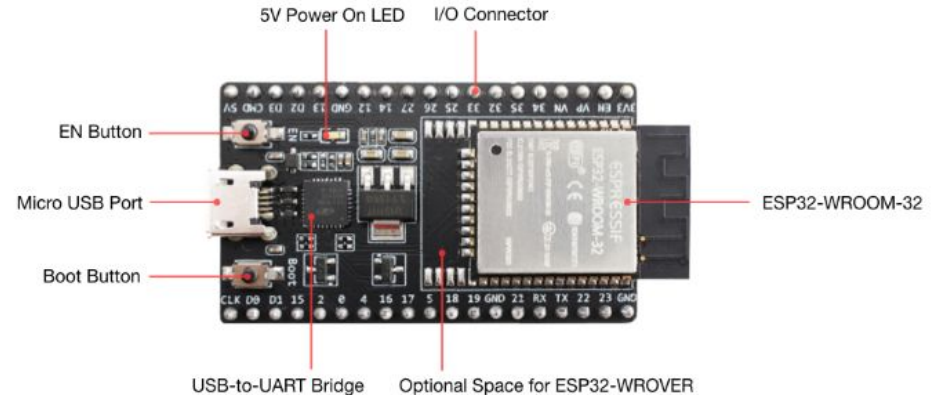
Autor	Microcontrolador	Sensor	Comunicação	Broker
Schoepping	WisBlock Core	TOF	LoRa	Tago.io
IEEE	-	Infravermelho	5G	-
Meu trabalho	ESP32	Ultrassônico	802.11	Mosquitto

Hardware

Jumpers	6
LEDs	2
Resistor	1
Protoboard	1
ESP32	1
Sensor HC-SR04	1
Sensor LJ18A3	1

Hardware

- **Protoboard:** Matriz de contatos.
- **ESP32:** Microcontrolador e módulo de conectividade
 - Conectividade Wi-Fi e Bluetooth
 - Processa dados
 - GPIOs



Hardware

- **Sensor HC-SR04:** Sensor ultrassônico para medir distâncias.
- **Sensor LJ18A3 :** Sensor indutivo para detectar presença ou aproximação de objetos metálicos.



Software

- **PlatformIO:** Plataforma utilizada para desenvolvimento de softwares para sistemas embarcados, integrado com VSCode.
- **Mosquitto:** Sistema de mensageria MQTT

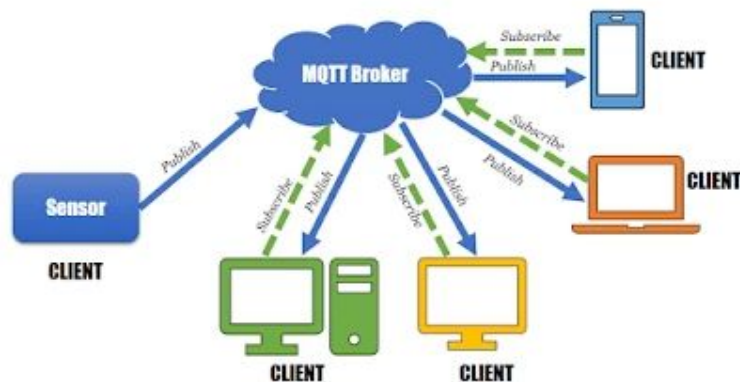
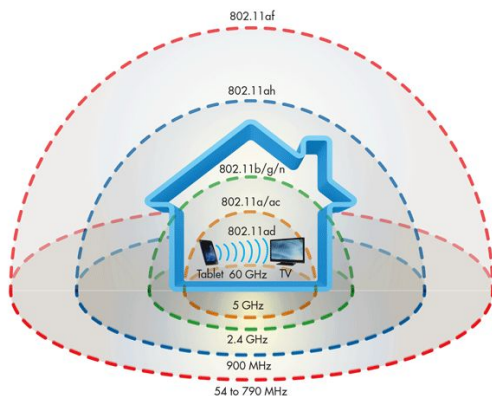


+

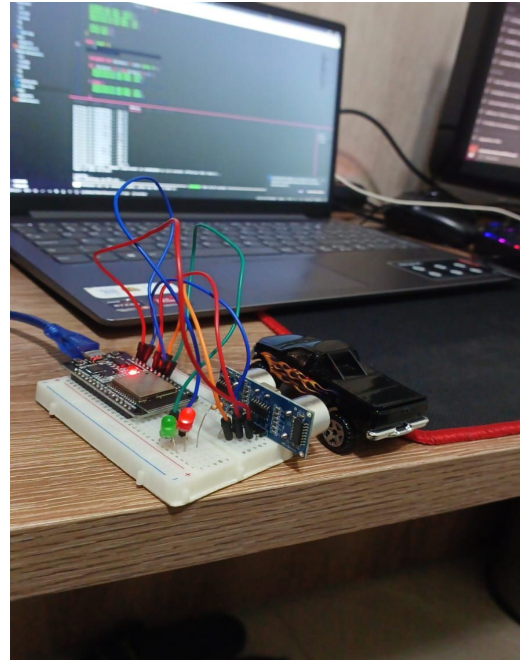
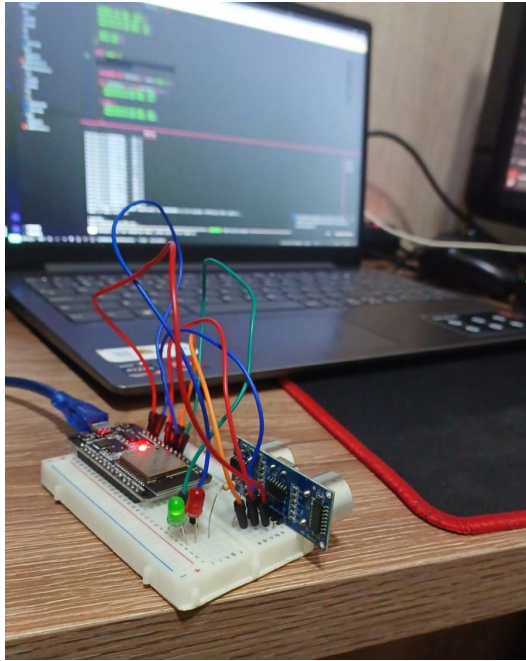


Tecnologias de Comunicação

- **802.11:** Diretrizes para a comunicação sem fio, unindo dispositivos em redes locais e garantindo a interoperabilidade em ambientes diversificados.
- **MQTT:** Comunicação assíncrona entre dispositivos em ambientes de Internet das Coisas.



Protótipo



Protótipo

- **Preparação do ambiente**
- Uso de bibliotecas NewPing, PubSubClient e Wifi.h
- Definição dos pinos utilizados para sensores e LEDs
- Credenciais de acesso para conexões Wi-Fi e MQTT.
- Função para publicação de dados no broker MQTT.
- Método para iniciar as conexões

Protótipo

```
void sendSensorDataToMosquitto(float distance) {  
    String payload = String(distance);  
    client.publish("topico_79384", payload.c_str());  
}
```

```
void setup() {  
    Serial.begin(9600);  
    pinMode(LED_PIN1, OUTPUT);  
    pinMode(LED_PIN2, OUTPUT);  
    digitalWrite(LED_PIN1, LOW);  
    digitalWrite(LED_PIN2, LOW);  
    WiFi.mode(WIFI_STA);  
  
    WiFi.begin(ssid, password);  
  
    while(WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
  
    Serial.println("Connected to WiFi");  
  
    client.setServer(mqtt_server, mqtt_port);  
    client.setCallback(callback);  
  
    while (!client.connected()) {  
        Serial.println("Attempting MQTT connection...");  
        if (client.connect("esp32_cli1")) {  
            Serial.println("Connected to MQTT broker");  
        } else {  
            Serial.print("Failed, rc=");  
            Serial.print(client.state());  
            Serial.println(" Retrying in 5 seconds...");  
            delay(5000);  
        }  
    }  
}
```

Protótipo

- **Funcionamento:**
- Leitura do Sensor Ultrassônico.
- Controle de LEDs com base nas distâncias medidas.
- Publicação dos dados no broker

Protótipo

- **Funcionamento:**
- Leitura do Sensor Ultrassônico.
- Controle de LEDs com base nas distâncias medidas.
- Publicação dos dados no broker

```
int init1 = 0;
uint32_t lastPublishTime = 0;

void loop() {
    if (!client.connected()) {
        reconnect();
    }

    if (init1 == 0) {
        client.publish("init", "init1");
        client.subscribe("topico_79384");
        init1++;
    }

    unsigned int distance = sonar.ping_cm();

    if (distance >= 0 && distance <= 8) {
        digitalWrite(LED_PIN1, HIGH);
        digitalWrite(LED_PIN2, LOW);
    } else {
        digitalWrite(LED_PIN1, LOW);
        digitalWrite(LED_PIN2, HIGH);
    }

    client.loop();

    if (millis() - lastPublishTime > 1000) {
        Serial.println("publish?");
        sendSensorDataToMosquitto(distance);
        lastPublishTime = millis();
    }

    delay(100);
}
```

Protótipo

- **Servidor Web:**
- Desenvolvimento de um servidor web em Node.js utilizando framework Express
- Utilização de Socket.io para estabelecer os sockets
- Biblioteca MQTT para configurar as publicações e inscrições com o Broker
- Inscrição no tópico onde o ESP32 está fazendo as publicações
- Tratamento de dados e utilização de página HTML para visualização do estado da vaga

Protótipo

```
const mqttClient = mqtt.connect('mqtt://192.168.1.20');

mqttClient.on('connect', () => {
  mqttClient.subscribe('topico_79384', (err) => {
    if (!err) {
      console.log('Conectado ao MQTT e inscrito em "topico_79384"');
    }
  });
});
```

```
mqttClient.on('message', (topic, message) => {
  console.log('Mensagem MQTT Recebida:', message.toString());

  const distance = parseFloat(message.toString());

  let status = 'Desconhecido';
  if (!isNaN(distance)) {
    status = distance < 9 ? 'Ocupado' : 'Livre';
  }

  io.emit('mqttMessage', { distance, status });
});
```

Protótipo

Sistema de detecção de vaga

Status da Vaga:



Vaga: Livre Distância: 37 cm

Sistema de detecção de vaga

Status da Vaga:



Vaga: Ocupado Distância: 3 cm

Análise dos Resultados

- Sensor HC-SR04 se mostrou eficaz para medir distâncias e indicar se vaga está ocupada.
- O sensor ultrassônico só indica distância para o objeto independente do material
- Sensor LJ18A3 tem capacidade de fazer essa distinção, porém a fonte utilizada não foi suficiente para alimentá-lo.
- Desafios de configuração do broker MQTT:
 - HiveMQ, não teve sucesso em fazer as publicações pelo ESP32.
 - Tago.io sucesso em fazer as publicações, porém sem sucesso para fazer inscrição com servidor web.
 - Mosquitto teve sucesso em ambos e fácil implementação.

Conclusão

- IoT é uma abordagem promissora para resolver problemas urbanos relacionados a estacionamentos.
- Dispositivo consegue com restrições identificar presença de veículos e comunicar a disponibilidade de vagas por meio de uma aplicação web.
- Sensor HC-SR04 não o mais indicado para o protótipo.
- Melhorias futuras envolvem utilização do sensor LJ18A3

Referências

BRAZ, Fernando. **Benefícios ao instalar Sistema de Vagas!** Disponível em:

<<https://www.linkedin.com/pulse/benef%C3%ADcios-ao-instalar-sistema-de-vagas-fernando-braz/?published=t>>. Acesso em: 15 out. 2023.

BRAZ, Fernando. **Quantas horas por ano você gasta procurando uma vaga de estacionamento?** Disponível em:

<<https://www.linkedin.com/pulse/quantas-horas-por-ano-voc%C3%AA-gasta-procurando-uma-vaga-de-fernando-braz/?originalSubdomain=pt>>. Acesso em: 15 out. 2023.

ESPRESSIF. **ESP-IDF Documentation: ESP32**. Disponível em:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>. Acesso em: 15 out. 2023

Repositório HC-SR04. GitHub. Disponível em: <https://github.com/d03n3rfr1tz3/HC-SR04>. Acesso em: 15 out. 2023

Referências

Proença, M. H. (2022). **Arquitetura iot para aplicação em smart campus.**

Schoepping, V. E. (2022). **Projeto de estacionamento inteligente utilizando parking iot sensor.**

Zahid, A., Mufti, N., Ullah, S., Nawaz, M. W., Sharif, A., Imran, M. A., Abbasi, Q. H., et al. (2023). **IoT-enabled vacant parking slot detection system using inkjet-printed RFID tags.** IEEE Sensors Journal, 23(7):7828–7835



Obrigado

Luís Eduardo Bertelli
Email: luiseduardobertelli1@gmail.com

Professora Janine Kniess
07/12/2023

**UDESC – Universidade do Estado de
Santa Catarina**